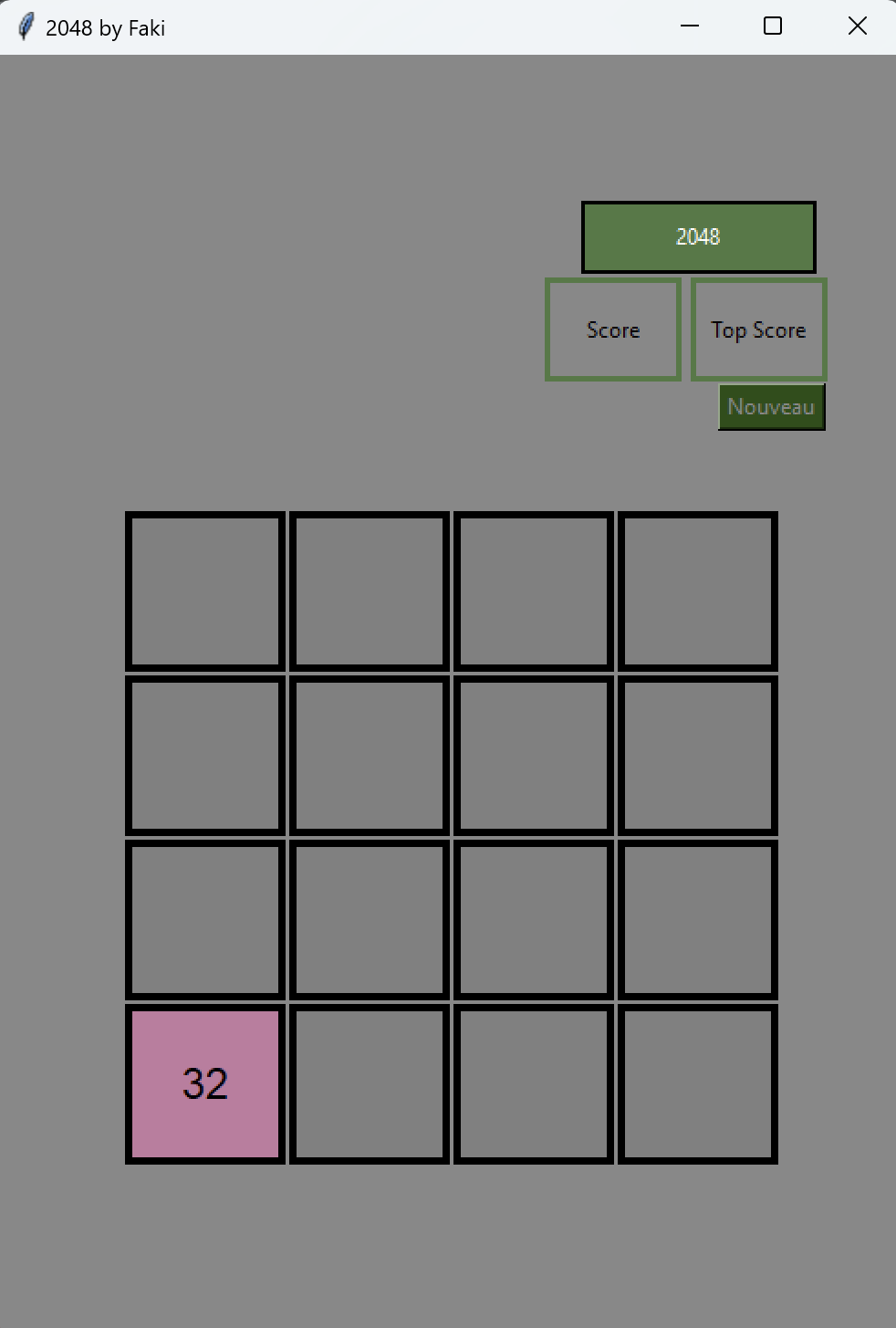


|  |
| --- |
| «2048 » |



Illustration

Ozdemir Fakime

Fakime-nur.ozdemir@cpnv.ch

Table des matières

SI-C1b

11.02.2023

[1 Introduction 3](#_Toc125451210)

[1.1 Cadre, description et motivation 3](#_Toc125451211)

[1.2 Objectifs 3](#_Toc125451212)

[1.3 Planification initiale 3](#_Toc125451213)

[2 Analyse 3](#_Toc125451214)

[2.1 Maquettes 3](#_Toc125451215)

[2.2 Stories / tests d’acceptation 3](#_Toc125451216)

[2.3 Stratégie de test 3](#_Toc125451217)

[3 Implémentation 3](#_Toc125451218)

[3.1 Vue d’ensemble 3](#_Toc125451219)

[3.2 Choix techniques 3](#_Toc125451220)

[3.3 Points techniques spécifiques 3](#_Toc125451221)

[3.3.1 Point 1 3](#_Toc125451222)

[3.3.2 Point 2 3](#_Toc125451223)

[3.3.3 Point … 3](#_Toc125451224)

[4 Tests 3](#_Toc125451225)

[4.1 Tests effectués 3](#_Toc125451226)

[4.2 Erreurs restantes 3](#_Toc125451227)

[5 Conclusions 3](#_Toc125451228)

[6 Annexes 3](#_Toc125451229)

[6.1 Sources – Bibliographie 3](#_Toc125451230)

[6.2 Journal de bord du projet 3](#_Toc125451231)

NOTE L’INTENTION DES UTILISATEURS DE CE CANEVAS:  
Toutes les parties en italique bleu (comme celle-ci) ne sont là que pour aider à comprendre ce qu’il faut mettre dans chaque partie du document.

**Vous veillerez donc à ce qu’il n’en reste aucune trace avant de rendre votre document final.**

De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n’aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l’alourdir inutilement.

**Pensez à changer le titre et le pied de page !**

# Introduction

## Cadre, description et motivation

Dans le cadre du CPNV, dans le module ICT-431, enseigné par Jean Philippe Chavey, je réalise la programmation du jeu 2048.

## Objectifs

Sprint 1 : Une maquette du jeu, un code avec deux exemples de tableaux/listes en mémoire. Des tuiles de toutes les valeurs de 2 à 8192 (=13 cases, de 21 é 213) avec leur couleur.

Sprint 2 : Un tableau de 4 valeurs « tassées » vers le début du tableau.

Mon objectif est de terminer toutes ces étapes et de livrer ce programme avec le moins d'erreurs possible. Et en même temps, voir mes erreurs et faire plus attention.

A la fin je veux avoir un jeu fluide et agréable à jouer pour toutes personnes, de bien réussir le jeu avec tout c’est mouvement.

## Planification initiale

Les étapes sont séparées en sprint. Les sprints ont chaqu’un plusieurs étapes.

Sprint 1 : du 10/02/2023

Sprint 2 : 10/03/2023

Sprint 3 : 24/05/2023

Défense 30-31 mars et 6 avril, démonstration devant la classe.

# Analyse

L’analyse détaille ce qui va être fait. A quoi va ressembler le produit fini. Comment il va fonctionner.

Elle doit faire l’objet d’une revue avec le client ; on s’assure que l’on a bien compris ce qu’il attend du projet.

## Maquettes

*Une image contenant table

Description générée automatiquement*

## Stories / tests d’acceptation

Les tests ont été réalisés sur IceScrum :

https://icescrum.cpnv.ch/p/MA202320O/#/planning/3124/sprint/3126/details

## Stratégie de test

Option de test sur iceScrum et moodle.

# Implémentation

## Vue d’ensemble

Cette section décrit comment le système à réaliser interagit avec son entourage, en termes :

* D’utilisateur(s) humain(s)
* D’utilisateur(s) logiciel(s) (clients d’une API, par exemple)
* De réseau
* De ressources externes

## Choix techniques

Les divers choix qui ont été faits pour la réalisation du mandat, en termes de :

* Matériel
* Systèmes d'exploitation
* Logiciels tiers (utilitaires, frameworks, navigateurs cible,…)

Pour chaque élément cité, on donnera une justification du choix et on fera la distinction entre ce qui concerne le travail de réalisation et ce qui concerne l’utilisation en production

## Points techniques spécifiques

Je veux vous expliquer la fonctionne qui sert a tasser nos chiffres :

Cette fonction était une fonction difficile pour moi car j'avais du mal à comprendre sa logique. Mais après avoir compris la logique, j'ai réalisé que ce n’était pas si compliqué.

\*\*Pour explique mieux j'ai été aidé par un informaticien turc.

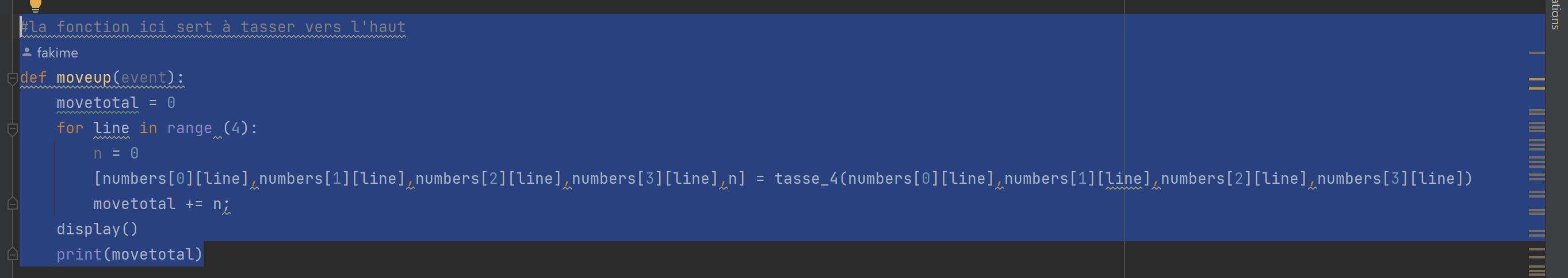
Cette fonction sert à déplacer les chiffres dans une "table" de 4 colonnes et 4 lignes vers le haut. Elle commence par initialiser une variable movetotal à 0, qui servira à compter le nombre de mouvements nécessaires pour déplacer les chiffres. Ensuite, elle parcourt chaque ligne de la table à l'aide de la boucle for.

Pour chaque ligne, la fonction appelle une autre fonction appelée tasse\_4 avec les quatre chiffres de la ligne en question. La fonction tasse\_4 sert à déplacer les chiffres vers le haut autant que possible tout en fusionnant les chiffres identiques. La fonction tasse\_4 renvoie la nouvelle ligne de chiffres ainsi que le nombre de mouvements effectués pour déplacer les chiffres.

La fonction moveup récupère la nouvelle ligne de chiffres retournée par tasse\_4 et l'assigne à l'emplacement approprié dans la table. Elle utilise également la variable n retournée par tasse\_4 pour ajouter au total de mouvements movetotal.

Enfin, la fonction moveup appelle une autre fonction appelée display pour afficher la table mise à jour et imprime le nombre total de mouvements effectués pour déplacer les chiffres.

En résumé, la fonction moveup sert à déplacer les chiffres d'une table de 4 colonnes et 4 lignes vers le haut tout en fusionnant les chiffres identiques, et elle affiche le nombre total de mouvements effectués pour y parvenir.



Ensuite je veux vous expliquer la fonctionne de tasse4 :

Cette fonction manipule ces nombres en les déplaçant vers la gauche et en fusionnant des nombres identiques. Elle renvoie ensuite une liste de 5 éléments contenant les nouveaux nombres ainsi que le nombre de déplacements nécessaires pour y parvenir.

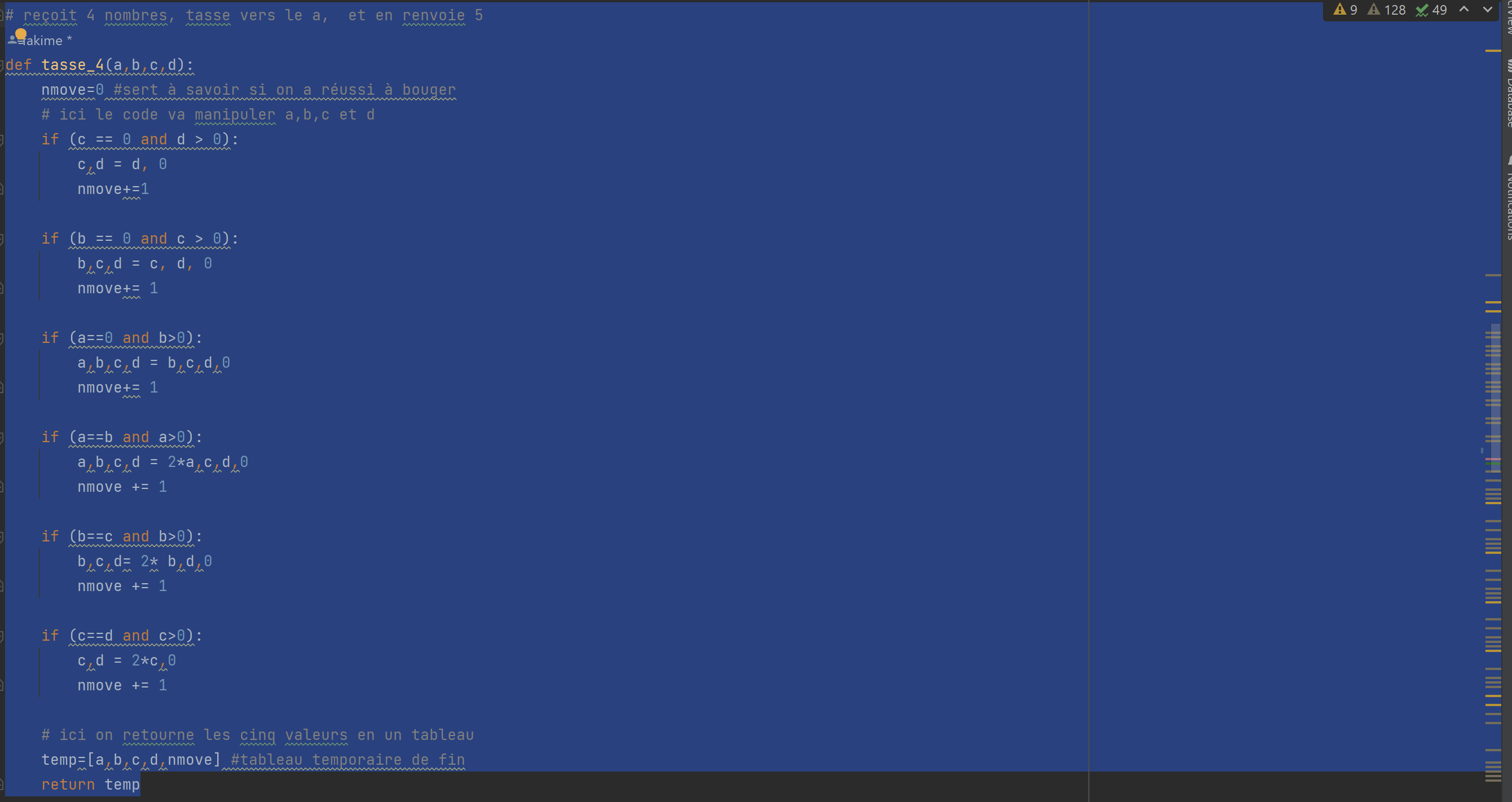
Le code commence par initialiser une variable nmove à 0, qui sera utilisée pour compter le nombre de mouvements nécessaires pour déplacer les nombres. Ensuite, il y a plusieurs blocs if qui déplacent les nombres en fonction de leur position actuelle dans l'ordre a, b, c et d.

Si le nombre c est 0 et d est plus grand que 0, le code déplace d dans la position de c et met 0 dans la position de d. De même, si b est 0 et c est plus grand que 0, le code déplace c dans la position de b, d dans la position de c, et met 0 dans la position de d.

Le code poursuit cette logique pour les nombres a et b, a et b, b et c, et c et d. Si deux nombres adjacents sont identiques, le code les fusionne en doublant le nombre et en mettant 0 dans la position de l'autre nombre.

Finalement, la fonction tasse\_4 renvoie les quatre nombres dans un tableau ainsi que la variable nmove qui contient le nombre total de mouvements effectués pour déplacer les nombres.

Et enfin, cette fonction sert à déplacer quatre nombres vers la gauche et à fusionner des nombres identiques, en renvoyant une liste de 5 éléments contenant les nouveaux nombres et le nombre total de mouvements effectués pour y parvenir.



### Point 1

### Point 2

### Point …

**Attention : Tout ce qui précède doit permettre à une autre personne de maintenir et modifier votre projet sans votre aide !**

# Tests

## Tests effectués

Référence à IceScrum dans lequel se trouve le résultat des tests.

## Erreurs restantes

S'il reste encore des erreurs:

* Description détaillée
* Conséquences sur l'utilisation du produit
* Actions envisagées ou possibles

# Conclusions

Développez en tous cas les points suivants:

* Objectifs atteints / non-atteints
* Comparaison entre ce qui avait prévu et ce qui s’est passé, en termes de planning et (éventuellement) de budget
* Points positifs / négatifs
* Difficultés particulières
* Suites possibles pour le projet (évolutions & améliorations)

# Annexes

## Sources – Bibliographie

Youtube : <https://www.youtube.com/watch?v=7EIvMsg5AWU&list=PL-Hkw4CrSVq9Y_RP7Q9Kn-bgZvVdl1cBy>

Courses : https://www.freecodecamp.org/learn/scientific-computing-with-python/

Sites Web utiles : https://python.doctor/page-apprendre-creer-fonction-en-python

## Journal de bord du projet

|  |  |
| --- | --- |
| **Date** | **Evénement** |
|  |  |

## Journal de travail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Jour** | **Semaine** | **Temps [h]** | **Type** | **Description** | **Remarques** |
| 01.02.2023 | 1 | 1,5h |  | Maquette |  |
| 03.02.2023 | 1 | 5h |  | Affichage |  |
| 08.02.2023 | 2 | 1,5 |  | Affichage |  |
| 10.02.2023 | 2 | 5h |  | Rendre maquette + projet avec le fonctions |  |
| 22.02.2023 | 3 | 1h |  | Avancement dans le programme |  |
| 24.02.2023 | 3 | 4h |  | Avancement dans le programme + iceScrum |  |
| 01.03.2023 | 4 | 1,5h |  | Avancement dans le programme + iceScrum + Github |  |
| 03.03.2023 | 4 | - |  | Absente |  |
| 07.03.2023 | 5 | - |  | Absente |  |
| 10.03.2023 | 5 | 5h |  | Fin sprint + Rndre |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |